情報処理学会論文誌 Vol. 49 No. 11 3725–3731 (Nov. 2008)

テクニカルノート

コンピュータ資源活用のための NQS パラメータ最適化における実験的検証

伊 藤 利 佳[†1]

本研究では，PC クラスタにおけるシステム資源の効率的利用のために，スケジューラ（NQS）のパラメータ設定の検証に対するアプローチを提案し，それを用いて行った実験結果を報告する．具体的には NQS と同じ振舞いをする NQS シミュレータを構築し，数理計画法をもちいて最適解を求める．これによって，これまで検証が不可能だったパラメータ設定の最適性の検証が可能となる．そこで構築したシミュレータを用いて行った数値実験を通して検証結果を報告する．

A Study for NQS Parameter Configuration to Improve the System Usage Efficiency

Rika Ito[†1]

In this paper, we propose an approach for the verification of the optimality of parameter settings of the scheduler (NQS) to improve the system usage efficiency in the PC cluster. We built NQS simulator based on the specification of Fujitsu NQS and find the optimal parameter settings with a mathematical method. We executed numerical experiments to verify the current parameter settings and report them.

1. はじめに

昨今では，ネットワーク性能の向上によって複数の PC を組み合わせた PC クラスタシステムが普及し，先進的なグリッドコンピューティング技術も実用に供されるようになってきた．このような分散システムの運用管理の立場からは，システム資源の利用効率の向上は 1 つの大きなテーマである．なぜならば，多数の資源を効率的に利用することはそもそも簡単なことではないうえ，利用効率の向上は，直接的にそのシステムの価値を底上げすることになるからである．

研究開発やエンジニアリング，金融機関におけるリスク管理業務を支援するシステムなどにおいては，分散システムのアーキテクチャが採用され，複数のユーザが同時に 1 つのシステムを利用する運用形態をとることが多い．これらの環境においては，逐次に投入されるジョブどうしの間に依存関係がほとんどないために，価格性能比から考えてスループット重視の分散システム環境が最適であるからである．

しかし，不特定多数のユーザによる共用システムであるため，資源配分の公平性の確保にも慎重に取り組まなければならない．ユーザが投入するジョブがなんの管理も受けなければ，ジョブどうしがシステム資源を奪い合い，利用効率に悪影響を及ぼすばかりでなく，ユーザの満足度も低下させる原因となるからである．

公平性に配慮しながらシステム資源の効率的利用を行うための技術の 1 つが，ジョブスケジューリングである．一般には，システム管理ソフトウエア群の 1 つの機能として提供され，スケジューラと呼ばれている．商用のスケジューラとしては，PBS Professional（Portable Batch System [⋆1]），Platform LSF（Load Sharing Facility [⋆2]），NQS（Network Queuing System [⋆3]）などがある．

独立行政法人理化学研究所（理研）においては，スーパコンピュータシステムとして，約 2000 CPU からなる「理研スーパーコンバインドクラスタ（RSCC）」が 2003 年 3 月から本格的に稼動した．これにともない，スケジューラとしては NQS を利用している．

スケジューラは，ユーザから投入されたジョブをいったん受け付け，優先順位やユーザごとに同時に実行できるジョブ数など，多数の設定値（パラメータ）をあらかじめ設定したルール（ポリシ）に基づき，ジョブを逐次に実行するというバッチ処理のシステムを構築する．割当て可能なシステム資源に余分がない場合，ユーザのジョブは待ち行列（キュー）に入ることになるが，ジョブスケジューラのポリシの設定によっては，空き資源があるにもかかわらず，ジョブが実行されずにキューに入ったままになるという状況が発生する．たとえ

†1 独立行政法人理化学研究所
RIKEN

⋆1 アルテアエンジニアリング株式会社
⋆2 プラットフォームコンピューティング株式会社
⋆3 富士通株式会社

ば，ユーザ 1 人あたりが同時に利用できる CPU 数に制限がある場合，自分の割当てを越してジョブを実行することはできないため，すでに限度いっぱいまで利用しているユーザのジョブは，資源全体として空きがあっても，キューに入ってしまい，実行されることはない．これは，資源の最大活用の観点から考えて問題である．この問題は，PC クラスタ，スーパコンピュータなど，システムの形態にかかわらず，バッチ処理を行うスケジューラを備えたすべてのコンピュータシステムが対象となりうる．そこで，本研究では，システムの資源活用の最適化を目的に，ジョブスケジューラのシミュレータを構築して数理計画法と組み合わせるという検証アプローチによって最適なパラメータ設定を求める数値実験を行ったのでその結果を報告する

2. 研究の意義

スケジューラの研究としては，スケジューリングに関する新たなアルゴリズムの提案[3),9)]，アルゴリズムに対する性能評価の研究がなされている．すなわち，既存のスケジューリングポリシに対し，性能を評価するための評価指標の定義および，定式化などを行う研究である[4),6)]．また，ジョブの実行時間の推測および，待ち時間短縮の試みなどもスケジューラの研究とともに研究されてきた[2),8)]．

昨今のコンピュータシステムの CPU 数の増加にともない，商用のジョブスケジューラは多くの機関において使用され，分野を問わず，さまざまなユーザに使いやすいものになってきている．

しかし，パラメータの設定に関してはどうだろうか．開発元からはスケジューラのパラメータの定義は提供されるが，パラメータの設定に関する情報はきわめて少ない．当然ではあるが，各利用施設の実運用環境に合わせて，どのように設定するべきかという情報はほとんど得られない．そのため，システム管理者は，過去の「勘と経験」に基づいて実運用システムの管理をせざるをえない．場合によっては，管理者は常時監視をし，空き資源があれば，パラメータの設定を変更して，強制的にジョブを流したりすることもある．しかし，これは明らかに非効率的である．

このような非効率性の要因としては「再現性のなさ」があげられる．ジョブの投入状況は逐次変化するため，同じ状況を作り出して検証を行うことができない．すなわち，別のパラメータ設定だった場合にはジョブがどのように実行されるか，ということが検証できないため，過去に設定したパラメータに対しても最適性を評価をすることができない．したがって，「より最適なパラメータ設定があったのではないか」ということを検討することも不可能である．システム資源に対する利用効率向上の重要性は叫ばれているが，既存の商用スケジューラのパラメータ設定に関して「いかにパラメータを設定するべきか」という議論はほとんどなされていない．しかし，最適なパラメータ設定で運用を行った場合における効率向上は無視してよいほど小さい値なのであろうか．そこで，本論文では，ジョブの実行状況を再現できるよう NQS のシミュレータを構築し，パラメータ設定の検証を可能とするとともに，最適設定を求め実運用との比較を行った．

具体的には，NQS と同じ振舞いをする NQS シミュレータを構築し，これによって，ジョブを実際に投入したり実行したりすることなしに各ジョブの実行状況が再現され，システムの利用状況，待ち時間などの情報が得られる．これを用いて数理計画アルゴリズムに基づき，資源全体を有効活用するような最適なパラメータ設定の究明を行い，検証を行った．この一連の検証アプローチは，NQS だけでなく，他のさまざまなスケジューラにおいても手法として適用可能であり，広く応用可能である．また，この手法の最大のメリットは，大きな再投資をすることなく，システムの利用効率の向上が期待できるという点にある．近年のシステムにおける CPU の増加傾向を考えると利用効率向上はますます大きな課題となっていくと考えられる．

3. RSCC のシステム概要と NQS

本章では，RSCC システムおよび NQS の概要について説明する．

3.1 RSCC のシステム環境

現在，理研においては，スーパコンピュータシステムとして，2048 CPU からなる RSCC が使用されている．システムの構成は，**表 1** のとおりである．システムは，PC1〜PC5 までの 5 つのクラスタに分割して運用されており，実験を行った PC1 にはシステム全体のうち，半分の 1024 CPU が割り当てられている．

3.2 PC1 におけるキュークラス

PC1 においては，現在設定されているキュークラスの数は 6 個である．並列度によって，32 CPU，64 CPU，128 CPU の 3 種類があり，それぞれが制限時間によって，ショート（10 時間）とロング（72 時間）の 2 種類に分かれているので，計 6 個のキュークラスが設定されている．s 032 など，以下の 6 個はキュークラスの名称である．

ショート：s 032，s 064，s 128
ロング　：l 032，l 064，l 128

表 1　RSCC システムの構成
Table 1　System components.

CPU	Intel Pentium Xeon 3.06GHz
Nodes	1024（2048CPU）
Peak Performance	12.4 TFLOPS
Memory	4GB/node，2GB/node
HDD	146GB/node
OS	Linux (Red Hat Version 8)
Scheduler	NQS(Fujitsu Version 1.0，NQS-JM)

3.3　NQS の概要

NQS は，アメリカ航空宇宙局（NASA）の航空力学数値シミュレーション計画の一環として開発されたジョブスケジューリングシステムを基に富士通株式会社が機能拡張して，Sun OS，VPP 環境で使用できるよう移植したものである．現在では，Solaris，Linux（Red hat 系）環境で動作しており，さまざまな大学や研究機関などで広く利用されている．

4.　NQS シミュレータの構築

研究を行うにあたり，ジョブのディスパッチアルゴリズムをシミュレートするような NQS シミュレータを構築した．これによって，実際にジョブを流すことなく仮想的に，パラメータ設定値に応じたジョブシーケンスを検証することができるようになった．本シミュレータのシミュレーションには，実運用の記録として残されているジョブシーケンスである過去の統計情報を利用する．仕様は**表 2** のとおりである．

4.1　NQS シミュレータにおけるパラメータ

NQS には，表 2 にあるように，RUN LIMIT，URUN などさまざまなパラメータが存在し，ジョブの実行をコントロールしている．たとえば，RUN LIMIT は，一般に同時実行多重度と呼ばれ，キューに対する制限を定義する．これは，同一キュー内で同時に実行できる最大のジョブ数を定義するパラメータである．また，URUN は，ユーザに対する制限を定義する．すなわち，同一ユーザが同一キュー内で同時に実行できる最大ジョブ数を規定するパラメータである．どちらのパラメータも，空き資源がある場合でもこの本数を超えてジョブを流すことはできない．これら以外にも，スケジューラには，さまざまなパラメータがあり，これらのパラメータの設定が，ジョブの実行を状況に応じて逐次に決定するポリシの役割を果たしているため，パラメータの設定は資源活用に大きく関わっている．

表 2　NQS シミュレータの仕様
Table 2　The specifications of NQS simulator.

仕様	NQS (Fujitsu)-Ver.1.0
	NQS-JM
パラメータ	PRIORITY RUN LIMIT, NEXT RUN
	URUN, ULIMIT, COMPLEX RUN LIMIT

表 3　シミュレータの入出力成分
Table 3　A part of past log data.

Input	・User ID ・Queue No. ・Number of CPU needed ・Submitted Time ・Elapse Time ・Parameter setting
Output	・Start Time ・Finish Time ・Waiting Time

4.2　NQS シミュレータ構築の概要

NQS シミュレータの構築のためのサンプル期間は 2006.6/20–7/10 とした．この期間は，システム管理者による強制的なジョブの投入がなく，NQS のアルゴリズムが統計情報にそのまま反映されている期間であった．このため，この期間のログデータをサンプルデータとして利用することによって，シミュレータの再現性の確認を行った．

その結果，当該期間におけるすべてのジョブの開始順序は過去のログデータと一致した．ただし，ジョブの開始時刻については，実際の運用においては，実行開始命令が出てから実行までに遅延が生じる場合があるため，シミュレーションの結果と比較した際にその分のタイムラグが生じるがこれはやむをえない．そのためこれは無視するものとすると，構築したシミュレータはスケジューラの振舞いをほぼ再現しているものと考える．

表 3 にシミュレータに対する入出力の成分を示す．過去の統計情報および，パラメータ設定からなる入力ファイルをシミュレータに入力すると，NQS のディスパッチアルゴリズムとパラメータに基づき，各ジョブがディスパッチされ，ジョブの実行開始時刻，終了時刻，待ち時間が出力として得られる．

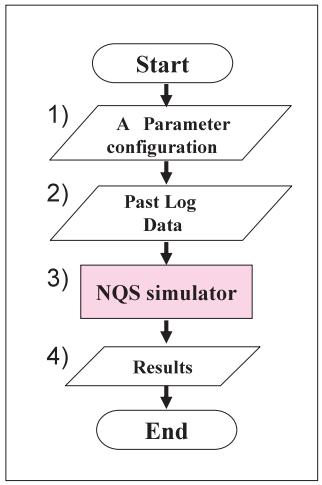


図 1　シミュレーションの流れ
Fig. 1　A procedure of simulation.

図 1 にシミュレータのフローチャートを示す.
1) RUN LIMIT などのスケジューラのパラメータ設定値を入力.
2) 過去の統計情報（ユーザ ID，キュー番号，使用 CPU 数など）を入力.
3) 過去の統計情報とパラメータ値，NQS のポリシに基づき，ジョブに資源が割り振られる.
4) 実行開始時刻，終了時刻，待ち時間などの最終結果を得る.

与えたパラメータ設定に応じてディスパッチの状況は変化するが，シミュレータによってさまざまなパラメータ設定による結果を取得することが可能である．このため，別のパラメータで図 1 の手続きを繰り返すことにより，最適なパラメータ設定を導くことができる．また，インプットする過去の統計情報を変更すると，同じパラメータ設定でもジョブの実行状況は異なるが，その場合もシミュレータによって，最適なパラメータ設定をプログラムにより導き出すことが可能となった．その結果，実運用における設定と最適設定とを比較検討することが可能となった．すなわち，これまでは，計算機上では，定量的に扱えなかったデータおよび，スケジューラの振舞いなどを数値として計算機上で扱うことができるようになり，各パラメータ設定に対する一定の評価が可能になった.

5. 数 値 実 験

今回，最適化の手法としてはヒューリスティックな解法として知られている Randomized Local Search をもちいた[1),7),10)]．本論文では詳細は省くが，この手法は，厳密な解を得ることが困難である場合，現実的な時間内で比較的良好な解を得る手法として，工学の現場では広く用いられている手法である．本手法を用いて最適解を探索した結果を以下で報告する.

5.1　最適化の評価基準

NQS のパラメータ設定を検証するために，数値実験を行って最適なパラメータを求めた．対象としたのは，2006 年 3 月期である．この時期は投入されるジョブが 1 年で最も多い時期であったためこの時期を選択した．実際の利用状況を実運用環境で再現することは不可能であるため，構築した NQS シミュレータを数理計画法の 1 つである Randomized Local Search に組み込むという形で実験を行った.

パラメータを評価する指標としてはさまざまな指標が考えられるが，本実験においては最大待ち時間を選択し，さらに付帯的な指標として，システム利用効率および平均待ち時間を選択した．システム利用効率は，ここでは SUE で表し，以下のように定義する（**図 2**）.

$SUE(\%) = S/V * 100$

V：全期間 [秒] $*$ 1024 (CPUs).

S：実行したジョブが要した CPU 数と時間の積分値.

すなわち，システム利用効率は，利用されたシステム資源をシステム全体の資源によって除した値を 100 分率で示した値である．これは，最適化による効率変化を判断するうえで有用であると考えられるので，付帯的な評価基準として加えた．以降においてはこの値をシステム利用効率と呼ぶことにする.

平均待ち時間は，期間内の待ち時間の総和（秒）を，処理したジョブ数で割ったものとする.

5.2　パラメータの最適化実験例

表 4 に 2006 年 3 月期の実験結果を示す.

表中の表記は以下のとおりである.
SUE：システム利用効率
MWT：最大待ち時間.
AWT：平均待ち時間

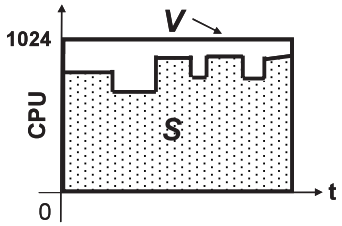


図 2　システム利用効率
Fig. 2　System usage efficiency.

表 4　2006 年 3 月における実験結果
Table 4　Results of the current and the optimal parameter.

Items	CCF	OPT	improve(%)
SUE (%)	92.870	96.210	3.340
MWT [sec]	806,728	562,574	30.26
AWT [sec]	90,852	69,192	23.84

CCF：実運用におけるパラメータ設定
OPT：最適なパラメータ設定

3 月期は，多くのジョブがキューに並んだ月であった．実運用の設定では，最大待ち時間が 806,728 秒になるジョブもあったが，最適設定では 244,154 秒（30.26%）短縮でき，平均待ち時間も 23.84%短縮できることが確認できた．また，システム利用効率は，3.34%向上することが分かった．そもそものシステム利用効率が 90%以上であることを考慮すると，最適化することによって得られる効果のインパクトは非常に大きいと判断できる．

図 3 は 2006 年 3 月の全ジョブの待ち時間のヒストグラムである．横軸には待ち時間が示されており，縦軸は度数である．最適化されたパラメータ設定では，サブミットしてから 5,000 秒以内に実行されるジョブの本数が増加し，50,000 秒以上待たされるジョブの本数が大幅に減少することが確認された．

図 4 は 2006 年 3 月における CPU の使用状況である．横軸は日付で，グラフは 1 日の平均使用 CPU 数を示している．縦軸は CPU 数である．このグラフから，最適パラメータ設定では，当該期間における CPU の使用数が全体的に高い値で維持されることが確認された．

そこで，他の時期についても検証実験を行った．2006 年 3 月期で求めた最適なパラメータ設定を，2005 年のログデータに用いて実験を行った．

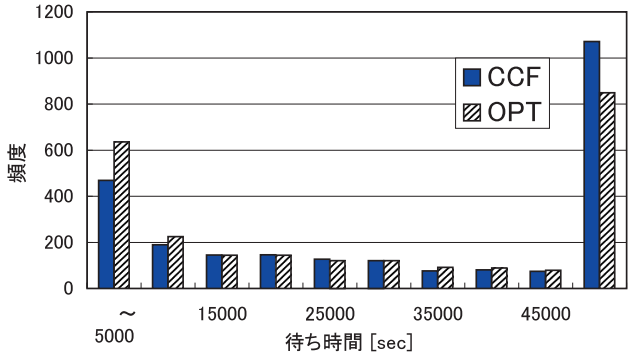


図 3　待ち時間の比較（2006/03）
Fig. 3　Histogram of the waiting time.

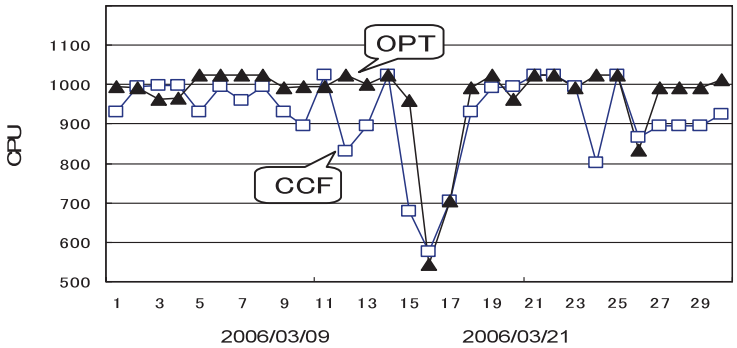


図 4　CPU 使用状況の比較（2006/03）
Fig. 4　Comparison of CPU usage.

2005 年 1 月から 12 月までのデータでシミュレートを行った結果を**表 5**，**表 6**，**表 7** に示す．システム利用効率（SUE）に関しては，改善率の小さい月も含めると，すべての月において改善がみられた．特に，3 月においては，5%以上もの改善がみられた．月によってばらつきはあるが，システム資源活用の観点から，1 年間全体で考えると無視できない値であると考える．

表 6 に最大待ち時間（MWT）の比較を示す．最大待ち時間に関してもすべての月において時間が短縮されることが確認された．特に 4 月，5 月においては大幅に時間が短縮される

表 5　2005 年における SUE の比較
Table 5　Results of SUE (%) in 2005.

SUE (%)			
2005	CCF	OPT	improve(%)
1	40.662	43.451	2.789
2	59.249	60.145	0.897
3	82.345	87.854	5.509
4	74.263	74.375	0.112
5	47.436	48.432	0.996
6	66.389	66.931	0.542
7	87.535	89.167	1.631
8	52.063	52.987	0.924
9	68.540	69.015	0.475
10	66.755	67.336	0.581
11	71.361	71.442	0.081
12	75.776	77.292	1.516

表 6　2005 年における MWT の比較
Table 6　Results of MWT [s] in 2005.

MWT [sec]			
2005	CCF	OPT	improve(%)
1	181,245	170,091	6.15
2	138,540	56,054	59.54
3	293,530	150,323	48.79
4	103,591	35,826	65.42
5	166,438	49,699	70.14
6	180,559	128,328	28.93
7	638,302	514,889	19.33
8	277,105	241,053	13.01
9	213,325	166,239	22.07
10	254,652	146,014	42.66
11	184,861	103,634	43.94
12	286,798	168,283	41.32

ことが示された．

また，表 7 に平均待ち時間の比較を示す．平均待ち時間に関しても，システム利用効率，最大待ち時間同様に，大幅な改善がみられ，全体的にすべての月において待ち時間が小さくなることが確認された．

表 7　2005 年における AWT の比較
Table 7　Results of AWT [s] in 2005.

AWT [sec]			
2005	CCF	OPT	improve(%)
1	1,932	1,207	37.52
2	2,515	938	62.70
3	12,747	8,189	35.76
4	1,860	877	52.88
5	3,589	769	78.56
6	45,140	38,887	13.85
7	94,892	76,698	19.17
8	36,319	22,644	37.65
9	30,187	19,446	35.58
10	23,810	14,334	39.80
11	13,894	9,597	30.93
12	45,952	31,292	31.90

6. ま と め

シミュレータを構築したことによって，本来は定量化できないスケジューラの振舞いを数値として扱うことができるようになり，計算機上で最適解を求めて比較することができるようになった．その結果，さまざまなパラメータ設定を比較検討することが可能となり，これによってパラメータ設定がシステム資源の有効活用に大きなインパクトを与えていることが改めて明らかになった．また，実運用のパラメータ設定においては，大幅な利用効率向上の余地があることが示された．さらに，システム利用効率の向上だけでなく，最大待ち時間や平均待ち時間の短縮化にもつながることから，シミュレータを用いた本検証アプローチは，システムを管理する施設側だけでなく，システムを利用するユーザにとっても大きなメリットがあるのではないかと考える．

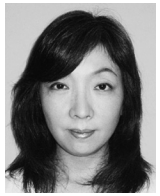
今後は実運用環境で利用可能にするよう，動的な最適化についての研究を行う．

謝辞　本研究にあたり，統計情報の取得，NQS の仕様の調査，データの収集などのために尽力をいただきました池田輝彦氏をはじめとし，お世話になった富士通株式会社の方々に深く御礼申しあげる．

参 考 文 献

1) Aarts, E. and Lenstra, J.K.: *Local Search in Combinatorial Optimization*, John Wiley & Sons (1997).
2) Downey, A.B.: Using Queue Time Predictions for Processor Allocation, *Proc. 3rd Workshop on Job Scheduling Strategies for Parallel Processing*, pp.35–57 (1997).
3) Dror, G.F.: Packing Schemes for Gang Scheduling, *Proc. 1st Workshop on Job Scheduling Strategies for Parallel Processing*, New York, pp.89–110 (1995).
4) England, D. and Weissman, J.B.: Costs and Benefits of Load Sharing in the Computational Grid, *Proc. 10th Workshop on Job Scheduling Strategies for Parallel Processing*, pp.160–175 (2004).
5) Grotschel, M., Lovaz, L. and Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*, Springer, New York, USA (1988).
6) Hamscher, V., Schwiegelshohn, U. and Yahyapour, A.R.: Evaluation of Job-Scheduling Strategies for Grid Computing, *Proc. 1st IEEE/ACM International Workshop on Grid Computing*, Lecture Notes in Computer Science, Vol.1971, pp.191–202 (2000).
7) Polak, E.: Optimization Algorithms and Consistent Approximations, *Applied Mathematical Science*, Vol.124, Springer-Verlag, New York (1997).
8) Tsafrir, D., Etsion, Y. and Dror, G.F.: Modeling User Runtime Estimates, *Proc. 11th Workshop on Job Scheduling Strategies for Parallel Processing*, pp.1–35 (2005).
9) Zhang, Y., Franke, H., Moreira, J. and Siva, A.S.: An integrated approach to parallel scheduling using gang scheduling, backfilling, and migration, *IEEE Trans. Parallel & Distributed, Systems*, Vol.14, No.3, pp.236–247 (2003).
10) 今野　浩，鈴木久敏：整数計画法と組合せ最適化，日科技連 (1982).

(平成 20 年 6 月 17 日受付)
(平成 20 年 9 月 10 日採録)



伊藤　利佳（正会員）

東京理科大学大学院工学研究科経営工学専攻博士課程修了．博士（工学）．理化学研究所情報基盤センター研究員．最適化を用いたデジタルフィルタの設計，ジョブスケジューラの最適化等最適化とその応用に関する研究に従事．オペレーションリサーチ学会，電子情報通信学会各会員．